

SICONOS Template #3

Simulation of a dry-friction oscillator with one discontinuity surface. Comparison between an event based scheme in Matlab and a time-stepping scheme in Scilab

V. Acary*, P. T. Piiroinen†

Version 1.0
April 14, 2006

1 Description of the physical problem: Dry-friction oscillator(Feigin)

Petri

2 Filippov systems : a general abstract class of NSDS.

In this section, we provide a short description of a general abstract class of Non Smooth Dynamical system (NSDS): Filippov systems. This type of system is a special case of Differential Inclusions (DI) given by

$$\dot{x} + g(x, t) \in F(x, t), \quad x \in \mathbb{R}^n, t \in I = [t_0, T] \subset \mathbb{R} \quad (1)$$

where :

- $x \in \mathbb{R}^n$ is the finite-dimensional state vector
- $g(x, t) : \mathbb{R}^n \times I \mapsto \mathbb{R}^n$ is a continuous function,
- $F(x, t) : \mathbb{R}^n \times I \mapsto \mathbb{R}^n$ is a set-valued mapping that is discontinuous on a number of hypersurfaces of the state space \mathbb{R}^n . The precise definition of this function is given below and characterize the Filippov system.

If the function $F(x, t)$ is reduced to the singleton $\{0\}$, then the equation 1 is a standard Ordinary Differential Equation. For Filippov system, the definition of the function $F(x, t)$ must respect the following definition. We start with a single hypersurfaces Σ which is called the switching boundary and is defined by the scalar function $h(x)$ such that :

$$\Sigma = \{x \mid h(x) = 0\}, \quad (2)$$

The hypersurface Σ splits the state space into two sets, S_+ and S_- , defined by :

$$S_+ = \{x \mid h(x) > 0\} \text{ and } S_- = \{x \mid h(x) < 0\} \quad (3)$$

Let us consider the (continuous) restriction of the function $F(x, t)$ on S_+ and S_- ,

$$\begin{cases} f_+(x, t) \triangleq F(x, t)|_{S_+} \\ f_-(x, t) \triangleq F(x, t)|_{S_-} \end{cases} \quad (4)$$

*INRIA Rhone-Alpes, vincent.acary@inrialpes.fr

†

The function F must be defined on Σ as the convex hull of all limits, i.e. :

$$F(x, t) = \bar{co} \{f_+(x, t), f_-(x, t)\} \quad (5)$$

This convex hull in the case of two functions f_+ and f_- may be stated as :

$$\bar{co} \{f_+(x, t), f_-(x, t)\} = \{(1 - \lambda)f_-(x, t) + \lambda f_+(x, t)\} \quad (6)$$

More generally, from a software architecture point of view, we will try, if possible, to present general abstract class of NSDS in a more general framework where the NSDS consists of:

1. a smooth dynamical system with boundary conditions in terms of state variables (§ 2.1) :

$$\dot{x} + g(x, t) = R \quad (7)$$

2. a set of input/output variables and theirs relations with the state variables (§ 2.2),

$$y = x \quad (8)$$

$$\lambda = R \quad (9)$$

3. a set of nonsmooth laws which rely on the input/output variable and the events (§ 2.3)

$$\lambda \in F(y, t) \quad (10)$$

2.1 Redaction note V. ACARY

Perhaps the definition of the non smooth law will be better in terms of state variable $y = h(x)$ and then the relation will be

$$y = h(x) \quad (11)$$

$$\lambda = R \quad (12)$$

together with the Non Smooth Law :

$$\lambda \in F(h^{-1}(x), t) \quad (13)$$

The reason why we write explicitly such a decomposition of the NSDS is :

- to be able to handle with every generic ODE as a Dynamical system and use generic numerical integration scheme
- to be able to change the relation type
- to be able to change the non smooth law and use a non law of type Filippov in other application.

2.1 Dynamical system and Boundary conditions

As we have made above, we can define the smooth dynamical part of the Filippov systems as :

$$\dot{x} + g(x, t) = R \quad (14)$$

Boundary conditions

The boundary conditions for an Initial Value Problem (IVP) are given as

$$t_0 \in \mathbb{R}, \quad x(t_0) = x_0 \in \mathbb{R}^n. \quad (15)$$

2.2 Relation between constrained variables and state variables

In a general way, the dynamical system is completed by a set of non smooth laws which do not concern directly the state vector. The set of such variables, denoted y , on which we apply the constraints depends, in a very general way, of the state vector x , such that

$$y = x. \quad (16)$$

In the same way, we can write :

$$\lambda = R \quad (17)$$

2.3 Definition of the Non Smooth Law between constrained variables

The Non Smooth law may be formulated as :

$$\lambda \in F(y, t) \quad (18)$$

with

$$F(y, t) \begin{cases} f_1(y), & y \in S_+, \\ f_s(y, \mu), & y \in \hat{\Sigma}, \\ f_2(y), & y \in S_-, \end{cases} \quad (19)$$

where

$$f_s(y) = \frac{f_+(y) + f_-(y)}{2} + \mu(y) \frac{f_-(y) - f_+(y)}{2}, \quad (20)$$

The resolution of this Non Smooth Law for one hyper-surface is completely and leads to :

$$\mu(y) = -\frac{\langle \nabla h(y), f_+(y) + f_-(y) \rangle}{\langle \nabla h(y), f_-(y) - f_+(y) \rangle}, \quad y \in \hat{\Sigma}. \quad (21)$$

2.4 Equivalence between Filippov system and Lagrangian dynamical system with dry Friction

In this section, we recall how a Lagrangian Dynamical System (see Template #1) may reformulated as a Filippov system. The equation of motion of a Lagrangian dynamical system type may be stated as

$$M(q)\ddot{q} + F(\dot{q}, q, t) = F_{ext}(\dot{q}, q, t) + R \quad (22)$$

where

- $q \in \mathbb{R}^n$ is the generalized coordinates vector. The dimension n of this vector is called the number of degree of freedom of the system. The first and the second time derivative of q , denoted \dot{q} and \ddot{q} , are usually called the velocity and the acceleration of the system.
- $\tau \in \mathbb{R}$ is the scaled time and introduced as a state variable.
- $M(q) : \mathbb{R}^n \mapsto \mathcal{M}^{n \times n}$ is the inertia term
- $F(\dot{q}, q, t) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$ is the internal force of the system,
- $F_{ext}(\dot{q}, q, t) : \mathbb{R} \mapsto \mathbb{R}^n$ is the external force.
- $R \in \mathbb{R}^n$ is the nonsmooth force due to the nonsmooth law in our case the Dry friction law

The dry friction law may be stated in terms of local variable, tangential velocity and tangential reaction to a surface. Let us assume that the tangential velocity is given by :

$$\dot{u}_T = H(q)\dot{q} \quad (23)$$

and the relation between the tangential friction force λ_T and R is given by :

$$R = H^T(q)\lambda_T \quad (24)$$

The Dry friction law may be written as :

$$-\lambda_T \in \partial\Phi_{C_g}(\dot{u}_T) \quad (25)$$

where C_g is the disk of \mathbb{R}^2 of radius g , the coefficient of friction. In two dimension, the disk is reduced to an interval of \mathbb{R} such that :

$$-\lambda_T \in \partial\Phi_{[-g,g]}(\dot{u}_T) \quad (26)$$

To reformulate a Lagrangian dynamical system into a Filippov system, a first general step is to denote the state of the system as

$$x = \begin{bmatrix} q \\ \dot{q} \\ \tau(t) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad (27)$$

which is a vector of dimension $2 \times n + 1$.

2.2 Redaction note V. ACARY

TO be finished ...

$$F_{ext}(\dot{q}, q, t) = \begin{cases} F_{ext1}(\dot{q}, q, t), & h(\dot{q}, q, t) > 0, \\ F_{ext2}(\dot{q}, q, t), & h(\dot{q}, q, t) < 0, \end{cases} \quad (28)$$

$h(\dot{q}, q, t) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}$ defines the discontinuity surface.

Introduce a surface Σ_{12} defined by the function $h(x) = h(\dot{q}, q, t)$ such that

which is defined as the surface where the external forcing F_{ext} is discontinuous. From this the sets

$$S_1 = \{x \mid h(x) > 0\} \text{ and } S_2 = \{x \mid h(x) < 0\} \quad (29)$$

can now easily be defined, in which the forcing F_{ext} is smooth.

The equation of motion (22) may be reformulated in terms of the state vector as

$$\dot{x} = \begin{cases} f_1(x), & x \in S_1, \\ f_s(x, \mu), & x \in \hat{\Sigma}_{12}, \\ f_2(x), & x \in S_2, \end{cases} \quad (30)$$

where

$$f_s(x) = \frac{f_1(x) + f_2(x)}{2} + \mu(x) \frac{f_2(x) - f_1(x)}{2}, \quad (31)$$

which is a often used formulation for ordinary differential equations of Filippov type. The set $\hat{\Sigma}_{12}$ is defined as the subset of Σ_{12} where the vector fields f_1 and f_2 are both pointing towards or away from the discontinuity surface Σ_{12} , i.e.

$$\hat{\Sigma}_{12} = \{x \in \Sigma_{12} \mid |\mu(x)| < 1\}. \quad (32)$$

3 The Formalization of the dry-friction oscillator problem into the class of Filippov NSDS

We assume that the system of a dry-friction oscillator belongs to the abstract class of the Filippov NSDS.

3.1 From the Lagrangian system to the Filippov system

From the input of the physical data, we give first the description of the data in terms of a Lagrangian system :

$$q \in \mathbb{R} \quad (33)$$

$$M(q) = 1 \quad (34)$$

$$F(q) = q \quad (35)$$

$$F_{ext}(q) = \sin(\omega t) \quad (36)$$

$$R \in \mathbb{R} \quad (37)$$

The tangential velocity and reaction are defined as follows :

$$\dot{u}_T = \dot{q} \quad (38)$$

$$R = \lambda_T \quad (39)$$

The Dry friction is define as follows :

$$\lambda_T = \begin{cases} -F_{fr}, & \text{sign}(\dot{u}_T) > 0 \\ +F_{fr}, & \text{sign}(\dot{u}_T) < 0 \end{cases} \quad (40)$$

and

$$\lambda_T \in [-F_{fr}; +F_{fr}], (\dot{u}_T) = 0 \quad (41)$$

The reformulation in terms of Filippov system leads to :

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \\ \omega t \end{bmatrix}, \quad (42)$$

$$g(x, t) = \begin{bmatrix} x_2 \\ x_1 - \sin(x_3) \\ \omega \end{bmatrix} \quad (43)$$

$$y = x \quad (44)$$

$$R = \lambda \in F(y, t) \quad (45)$$

$$h(y) = y_2, \quad (46)$$

$$F(y, t) = \begin{bmatrix} 0 \\ F_2(x, t) \\ 0 \end{bmatrix} \text{ with } F_2(y, t) = \begin{cases} -F_{fr}, & h(y) > 0 \\ +F_{fr}, & h(y) < 0 \\ [-F_{fr}, +F_{fr}], & h(y) = 0 \end{cases} \quad (47)$$

3.2 Dynamical system and boundary condition

see above

3.3 relations

see above

3.4 Non Smooth laws

see above

4 Description of the numerical strategy: Simulation using an event-driven scheme in Matlab

4.1 Time discretization of the dynamical system

In this section we provide an idea how to use an event driven scheme to solve Filippov type problems (in Matlab).

4.1.a Solvers and event detection

Nonlinear systems

For nonlinear dynamical systems any of the built-in adaptive integrators, such as `ode45` (4th order Runge-Kutta for nonstiff systems) or `ode15s` (Uses backward differential formulas (BDFs) also known as Gear's method for stiff systems), can be used to solve the smooth dynamics. It is of course possible to write your own code for solving ODEs, which should be straight forward.

For nonlinear dynamical systems the built-in event detection routines used with the integrators can be used to locate $y_1 = 0$. For a more grazing sensitive solver one can also look for $\dot{y}_i = \nabla h(x)f_i(x) = 0$, $i = 1, 2$. If a user provided solver is used then the user also has include an event detection scheme. For this, e.g. standard bisection, secant, or Broyden's method (a variant of the secant method) can be used and with a specified tolerance.

Linear system

For linear dynamical systems any of the built in integrators (as for the nonlinear systems) can be used to solve the smooth dynamics. However, it is also possible to use the `expm` function which is the matrix exponential or in very simple cases the actual solution can be given to the program for.

The same approach on event detection holds for linear systems as for nonlinear systems (see above).

Chatter and sticking/sliding

In impacting systems chatter and sticking are very common phenomena that often causes a lot of problems in event driven methods. To have sticking there are three condition that has to be fulfilled at impact. They are

$$y = 0 \text{ (impact condition)} \quad (48)$$

$$\dot{y} = 0 \text{ (grazing condition)} \quad (49)$$

$$\ddot{y} \leq 0 \text{ (acceleration condition)} \quad (50)$$

Sticking can be caused by either if the coefficient of restitution is zero or as the result of a chatter sequence. In chatter the system recognises infinite number of impacts in finite time. Obviously it is impossible to numerically solve for an infinite number of events, so the most common way to solve this is to set $\dot{y} = 0$ as the conditions (48) and (49) are fulfilled and

$$|\dot{y}| < C, \quad (51)$$

where C is a small and positive constant.

When the system is in sticking mode the condition to look for is **not** $y = 0$, since this is already fulfilled, but instead we look for $\ddot{y} = 0$ which is exactly when the acceleration is changing sign and eventually the direction of the velocity away from the impacting surface.

4.1.b Detection of and action at the impacting surface

In order to apply Newton's impact law the surface given by

$$y = h(x) = 0 \quad (52)$$

and possibly also

$$\dot{y} = \nabla h(x)f(x) = 0 \quad (53)$$

has to be located with an appropriate method, as describe above in Sec. 4.1.a.

During the ODE integration the event detector will eventually find the exact time and position of an impact, i.e. when $y = 0$. At such a point the integrator stops, wherafter the impact law is applied and the integrator is again started, but now with the new inital conditions given by the impact law.

Chatter and sticking/sliding

When the motion fulfills (48), (50), and (51) the solver stops and \dot{y} is set to 0 and the integration is restarted but now only looking for the event $\dot{y} = 0$. To avoid drift away from the surface due to numerical errors tt is possible to make the surface attracting by adding the term

$$\nabla Dh(x)^T h(x), \quad D > 0 \quad (54)$$

to the vector field.

Another and a more accurate idea is to use a local mapping att chattering. Such methods have been developed but not yet tested or included in a code.

4.2 Numerical Strategy

The numerical stratgey is given by the pseudo-algorithm 4.2.

5 Description of an other numerical strategy: Simulation using an Time-stepping scheme in Scilab

Vincent

6 Exploitation of the results

For the case of Lagrangian NSDS, the state variable and the reaction force vs. time are the basic results that we want to export out of the algorithm. Various of energies may be also expected by the user.

7 First analysis for the architectural point of view

To be defined ...